Lighting Design: A Goal Based Approach Using Optimisation

António Cardoso Costa[I II]
acc@dei.isep.ipp.pt

António Augusto Sousa[III II]
augusto.sousa@inescn.pt

Fernando Nunes Ferreira[III]
fnf@fe.up.pt

Abstract

There is a need for reliable lighting design applications because available tools are limited and inappropriate for interactive or creative use. Architects and lighting designers need those applications to define, predict, test and validate lighting solutions for their problems. We present a new approach to the lighting design problem based on a methodology that includes the geometry of the scene, the properties of materials and the design goals. It is possible to obtain luminaire characteristics or other kind of results that maximise the attainment of the design goals, which may include different types of constraints or objectives (lighting, geometrical or others). The main goal, in our approach, is to improve the lighting design cycle. In this work we discuss the use of optimisation in lighting design, describe the implementation of the methodology, present real-world based examples and analyse in detail some of the complex technical problems associated and speculate on how to overcome them.

Key words: lighting design, inverse design, optimisation, global illumination, light transport.

1. Introduction

In a 3D space for which we know the geometry and materials within, where should we place luminaires and what characteristics should they have to satisfy the lighting design goals?

This question hides very complex and challenging problems. The people that need answers to this question — architects, engineers and designers — would benefit very much if tools were available to define, predict, test and validate their lighting design solutions; instead, existing tools are limited in scope and inappropriate to interactive or creative use. Computer tools are capable of providing more detailed analyses than real or small-scale models. Designers are increasingly computerised and seek prediction and analysis tools for their problems. In this paper we describe and analyse a new approach that solves the problem of finding lighting solutions from the geometry, the materials and the design goals (an inverse approach). The calculation engine is a global illumination simulation program that uses the geometry, the materials and the light sources (as in a direct approach). The inverse approach is more complex because there can be many incompatibilities between the input information. These incompatibilities have a physical meaning and can be dealt with by changing some of the input data (geometry, material properties or design goals). This inverse approach also allows the exploration of lighting design solutions in virtual or real spaces. Fig. 1 shows our approach, where the designer includes design goals in the input data. A

[I] Departamento de Engenharia Informática do ISEP-IPP, Rua S. Tomé, 4200 Porto, Portugal.
[II] INESC Porto, Praça da República 93 R/C, Apartado 4433, 4007 Porto CODEX, Portugal.
[III] Faculdade de Eng. da Univ. do Porto, Rua dos Bragas, 4099 Porto CODEX, Portugal.

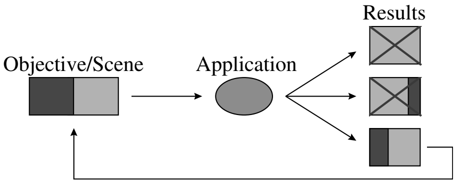

Fig. 1. Our Approach in Lighting Design

user-driven computational search iterates until some preset termination criteria is achieved. The best solutions are shown and a decision can then be made. The three types of solutions in the right part of Fig. 1 show that it is possible to generate impossible, unwanted or adequate solutions. The designer can choose to refine the design goals and restart the process or, alternatively, to apply his own judgement and choose the most suited solution available. The abstract description in Fig. 1. hides many difficult and resource consuming problems, for which available solutions are yet very unsatisfactory. The main goal of this work is to qualitatively improve the lighting design cycle. Improvement is achieved by letting the user include the design goals in the search for solutions and explore the different solution paths. If this is accompanied by technical improvements, then we would also attain another important goal: shortening the design cycle. The "Related Work" section describes other research work, while the "Overview" section explains our approach and methodology. The "Implementation" section describes the algorithm implemented and methods/techniques used. The "Results" section presents examples, problems found, technical solutions developed and some performance analyses. In the "Open Questions" section we discuss questions related to lighting design and our approach. Finally, conclusions are presented in the last section.

2. Related Work

In [Costa99], within the context of the proposed approach, the authors describe the methodology, the algorithm and an initial implementation, but without going into details. Schoeneman *et al.* [Schoe93] and Kawai *et al.* [Kawai93] tried to address the inverse lighting design problem, but solved it only partially for mostly diffuse environments and seem to have ignored many types of design constraints. Both used a radiosity-based algorithm [Cohen93]. Poulin *et al.* [Pouli97] described an approach that allows the interactive positioning of simplified luminaires from sketches of shadows, umbra, penumbra or highlights (as design constraints), but their approach does not seem capable of handling indirectly lit spaces. The Design Gallery approach of Marks *et al.* [Marks97] is also interesting, because it presents a new methodology for the exploration of solutions in large multidimensional problems, different from conventional methodologies like interactive evolution or inverse design. Results may be questionable because it seems they used a local illumination algorithm. Also, their dispersion phase (solution generation and filtering) can become prohibitively expensive in computation time if many solutions have to be generated for the following reduction (browse and select) phase.

3. Overview

In [Costa99] an overview of our approach was presented which explained the concept of an "ideal" lighting design approach – the inputs are:

- the geometry of the scene
- the properties of the materials
- a set of statements about lighting design goals.

The outputs are:

- a set of solutions (possibly empty)
- suggestions to overcome the limitations found.

Due to the complexity of the problem and the insufficiency of current technological resources, alternative "practical" approaches must be found that allow the implementation of computerised solutions with reasonable resource consumption – storage, time, etc. This work describes one "practical" approach and the algorithm, methods and techniques that we have found appropriate for its implementation. Lighting is a very complex mix of physical phenomena [Feynm85], but for most real world environments, some of those phenomena are very rare and simplifications can be made that allow simpler theories and models to be derived. Some of the simplifications that were made for developing a feasible implementation are:

- geometric optics – assumed (particle theory of light)
- media – non participating media
- *BSDF*[1] – symmetrical (no change if incoming and outgoing rays are switched)

Although the *BRDF* (reflected components of the *BSDF*) of real surfaces is symmetric, the *BTDF* (transmitted components) seldom is and that may invalidate the reversibility of lighting calculations. One way to circumvent this is to use the *basic radiance* concept of [Veach97]. To provide lighting data for our implementation it is important to use a lighting calculation tool with at least the following requisites:

- physically valid symmetric *BSDF* model (*BRDF* only if no transmission exists)
- reversibility of lighting calculations
- no restrictions for the properties of surfaces
- light sources with arbitrary radiance distributions
- radiance calculation anywhere in space.

Lighting design goals are mostly modelled by fictitious luminaires that are artificially introduced in the scene to provide means for computing solutions (this is the reason why reversibility of lighting calculations is so important). These luminaires may be:

- PL – previous luminaire; a luminaire in the scene, ie, a design condition
- IL – inverse luminaire; a fictitious luminaire used as a design goal
- DL – desired luminaire; a lighting design result.

3.1 Methodology

In some cases, lighting design problems are based on scenes that have some initial radiance distribution: predefined luminaires, daylighting, etc – this initial illumination is represented by PLs. After the designer quantifies the lighting design goals using ILs, an initial step is performed to account for the effect of the initial radiance distribution in the scene. This lighting simulation computes the effect of the PLs in the ILs. If those effects do not lead to contradictory situations, then the ILs are decomposed into a new set of ILs that account for the initial scene illumination. Using the ILs as radiance emitters, we must find a way of computing the incoming radiance in selected points and directions in the scene's volume and correlate that with a particular design solution. With an optimisation algorithm it is possible to calculate automatically solutions that maximise some function of that incoming radiance, if we are able to define

---

[1] Bi-directional Scattering Distribution Function.

such a function – this will be a cost function $F(x)$ of the optimisation phase, whose main goal is to find the global maximum. In most lighting design problems, a solution may be represented by different sets of parameters: luminaire position, luminaire spacing or almost anything relevant to the scene. Usually, the n parameters can be converted from real to suitable integer values, which makes the configuration parameter space a large set of n-dimensional points. Exhaustive searches seldom are an efficient way to find the solutions. We must then resort to local search strategies [Pirlo96], which basically move from one solution to another one in its neighbourhood according to some well defined rules. In lighting design, straightforward strategies are not adequate to solve the problem because the solution may be very difficult to find if the cost function is complex. Simulated Annealing (SA) [Kirkp84] is a suitable strategy for difficult optimisation problems and is able to process cost functions with quite arbitrary degrees of non-linearities, discontinuities and stochasticity, arbitrary boundary conditions and constraints imposed on the cost function and is statistically guaranteed to find the optimal solution [Ingbe93]. The non-linearities or discontinuities of the cost function may invalidate the use of optimisation methods based on gradient calculations, but with a continuous and "smooth" cost function, a two step technique joining SA and a gradient-based search method might prove beneficial.

4. Implementation

Fig. 2 shows an outline of the our algorithm. The input data is the geometry of the scene, the properties of materials and the lighting design goals. The geometry and properties are usually described in a quantitative way using conventional data formats. PLs and ILs describe initial scene's luminaires and lighting goals. The double arrow in Fig. 2 represents the communication channel between the optimisation and the lighting calculation tools. We have chosen the *Radiance* computer program for lighting calculations because it has been photometrically and photorealistically validated [Ward94]. The lighting calculations performed inside the optimisation loop of Fig. 2 are the main responsible for the large computation times in our implementation. Each calculation is a light transport computation in a set of directions around a point. Radiance behaviour around a point is calculated by collecting radiance samples around that point (SRD – spherical radiance distribution). Due to the complexity of lighting, many samples are necessary. In our test cases, we have found that 2048 "well distributed" samples on the direction sphere are a good compromise between accuracy and performance, although in many cases 1024

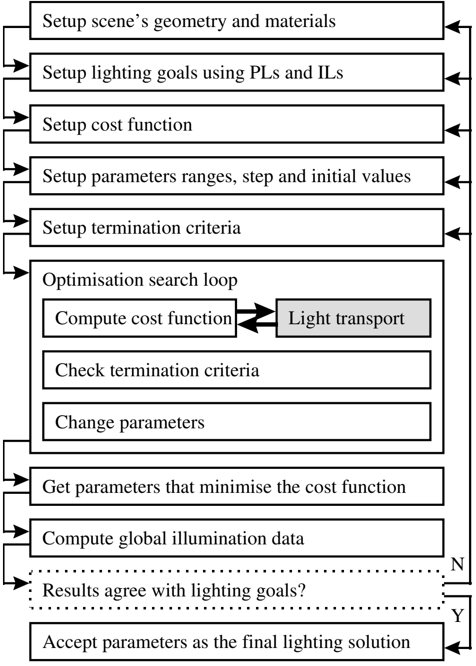

Fig. 2. Algorithm for our methodology

samples also work fine. We have also chosen the *ASA* [Ingbe1998] optimisation software (a SA algorithm). In all tested cases, it has produced good and reproducible results in less than 5000 iterations.

4.1 Design Goals and Cost Functions

The double arrow in Fig. 2 links the optimisation and the lighting calculation tools. The optimisation tool is basically a loop that tests different sets of configuration parameters while trying to maximise a cost function dependent on those parameters. To compute a cost function value, the parameters are passed to a user-defined script that manipulates them and calls the lighting calculation tool as necessary. This script (a representation of the cost function) is user definable and may include relationships, constraints, etc, between parameters and lighting data. Typically, the cost function script $F_{script}(x)$ will measure radiance arriving at particular points and produce a scalar value that is dependent on all the design goals included in this function. We have developed a script language for $F_{script}(x)$: the optimisation parameters are input parameters (x_1, x_2, etc) and it returns as result the calculation of the cost function value. This script language is like a programming language with operators (geometrical, logical, etc), flow instructions (choice, loop, etc) and is implemented with standard tools like GNU *flex* and *bison*. To compute radiance values arriving at a point within a solid angle, there is a special function **Importance**() that actually calls the lighting calculation tool to do it. If luminaire candlepower data is available and essential to the lighting design, then it can be used when computing the radiance values by scaling them with the appropriate candlepower factor (the candlepower distribution must be tabulated or defined analytically). The input parameters represent any design variable relevant to the lighting design problem that must be optimised. As such, they may represent spatial co-ordinates, angles, distances, etc. If a designer says *"I would like to enhance the lighting of this room by adding a projector type luminaire hanging from the ceiling so that illumination on the top of this table improves, but without glare effects or excessive lighting in the user face"*, this lighting design problem can be defined by the use of 2 ILs, one for the table top (positive) and another for the user face (negative). The cost function script must verify if the geometrical and directional constraints are respected and compute the fictitious radiance (emitted by ILs) arriving at points dependent on the input parameters to produce its final cost value. In this example, the input parameters represent the luminaire position and direction (5 parameters). The cost script in Code 1. represents those lighting design goals:

```
1.  # (x1,x2,x3):position (x4,x5):direction
2.  # other variables have user defined values
3.  V=Vector(FaceCenter,(x1,x2,x3))
4.  if Angle(FacePerp,V)<Vthreshold and
5.     Angle(FacePerp,Dir(-x4,-x5))<Athreshold return FAILURE
6.  WIL1/DL1=Importance(SceneIL1,x1,x2,x3,x4,x5)
7.  WIL2/DL1=Importance(SceneIL2,x1,x2,x3,x4,x5)
8.  return -(K1*WIL1/DL1-K2*WIL2/DL1)
```

Code 1. Cost function

- *line 4* - avoids the luminaire inside the user view
- *line 5* - avoids direct lighting on the user face
- *line 6* - measures the importance of table top IL_1
- *line 7* - measures the importance of user face IL_2
- *line 8* - computes the weighted effect of both ILs.

The contributions of the two ILs have opposite signs in *line 8* because IL_1 represents an objective to maximise while the IL_2 represents an objective to minimise.

5. Results

Radiance has a useful and reliable cache technique that accelerates the calculation of diffuse inter-reflections. In the initial light transport calculations, *Radiance* has to compute a lot of information to build the diffuse inter-reflection cache; after that, it is able to produce light transport data much more quickly, because most of the required information is cached. When working with complex geometries, it is also advantageous to use *Radiance* in persistent mode (it reads the input data and then executes ray-tracing commands when requested without being restarted in each design iteration). This way *Radiance* saves a lot of time setting up its internal data structures again and again. All the simulations were performed on PC's running *Linux* with recent versions of *Radiance* (3R1) and *ASA* (17.19) software. Each simulation was repeated at least 5 times and the best result was taken as the final result. All positional parameters were converted to integer 5cm units and all directional parameters were converted to integer 2° units to reduce the parameter search space to a reasonable size (but still too big to do exhaustive searches).

5.1 Test Case #1

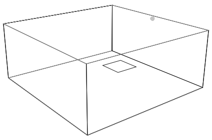

Fig. 3. Geometry of simple scene

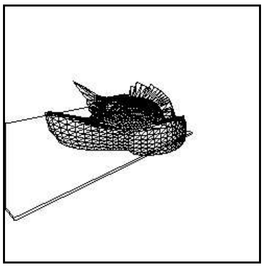

Fig. 4. SRD - PL

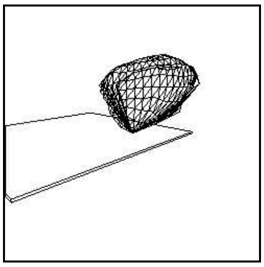

Fig. 5. SRD - IL

The scene is a simple room (6x6x2.5m) with an initial luminaire (PL), located in the right wall near the ceiling, as shown in Fig. 3. The table top is at height 0.8m above the floor and it is not centred in the room. The materials are slightly specular. The lighting design goal is to add another projector type luminaire near the ceiling (above 2.4m) so that the lighting on the table top becomes homogeneous. This lighting goal is modelled with a positive IL on the table top. To account for the PL lighting effects on the table top, it is necessary to calculate the initial lighting distribution over that table top and decide whether or not to subdivide the initial IL into smaller ILs. Fig. 4 shows the radiance distribution due to the PL in a point on the table top. This SRD shows how radiance arrives at the specified point (the larger lobes represent greater incoming radiance in the corresponding direction). If the designer wants a uniform radiance distribution around the point, then there will be some directions where the incoming radiance may already be greater than the desired threshold. For those directions there is no need to shoot fictitious radiance into the scene. For other directions where the desired radiance is not yet achieved, the missing radiance is emitted into the scene as fictitious radiance and it will be used to search for the best luminaire characteristics. The radiance distribution in Fig. 5 is a SRD representing a set of directions where the desired incoming radiance is not yet achieved. In this example, we subdivided the initial IL into four ILs to accurately account for the initial lighting on the table top. To add one projector type luminaire (DL) to the scene, the cost function can be the sum of incoming fictitious radiance emitted by the four ILs. We can anticipate that the solution should be symmetric

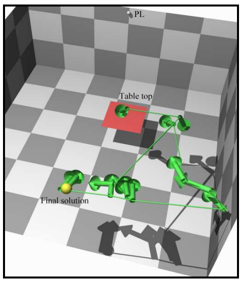

Fig. 6. Solution evolution

to the PL, near the ceiling in the opposite wall. The cost function script will call the lighting calculation tool four times (one for each IL) to compute a weighted sum of incoming radiances for each candidate point and solid angle. The image in Fig. 6 shows the evolution of solutions during the optimisation (arrows represent successive intermediate solutions and the final solution is shown with a small sphere). It can be noticed that initial solutions are near the table top, but the directional ILs make the intermediate and final solutions settle in a high region near the left wall. For this design study, because the geometry is very simple, each simulation produces good results in less than 2000 iterations. One simulation with 2000 iterations runs in less than 1 hour.

5.2 Test Case #2

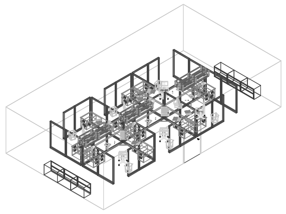

Fig. 7. Geometry of NRC office scene

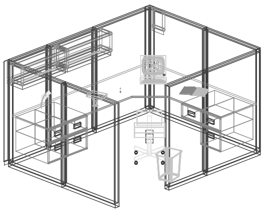

Fig. 8. Working cell in NRC office

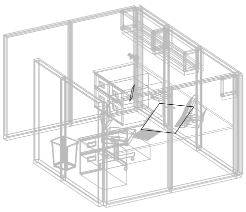

Fig. 9. ILs inside NRC working cell

The previous test case had a very simple geometry, but to show the advantages of this methodology a more complex and realistic scene is used. The next test case uses a scene that is based in a real office (Fig. 7). This scene has 10550 geometric objects (mainly polygons) and some of the materials are specular. The office is composed of 6 working cells and corridors. Each working cell is surrounded by vertical panels and has a L-shaped desk top, a computer screen, some drawers (below and above the desk top) and other small objects over the desk top (Fig. 8). An empirical lighting design solution is shown in Fig. 11 (see Appendix), where the solution is composed of 10 (2x5 grid) rectangular flat panel luminaires in the ceiling (ambient lighting) and 12 projector luminaires on the desk tops (task lighting). This design solution does not take into consideration some important design factors:

- best placement of luminaires for lighting
- glare in user faces for standard sitting positions
- shadow boundaries on the panels and desk tops
- redundancy in the amount of luminaires.

It would be desirable to have a lighting design solution where those factors could be accounted for during the design study, not in the end. The first step to achieve this is to define appropriate lighting design goals:

- lighting on the desk tops – positive IL
- avoid glare in user faces – negative IL
- luminaires in volumes – constraint
- luminaires in row – constraint.

The IL for each desk top has been defined as a rectangle in front of the computer screen and the IL for each user face is a small rectangle corresponding to a human face in standard sitting position – these ILs are represented by the black rectangles in Fig. 9. Let us suppose that the designer wanted to try several types of luminaires and arrangements.

5.2.1 Luminaire Type #1

```
1.  # ($x_1,x_2,x_3,x_4,x_5,x_6$),($x_7,x_8,x_9,x_{10},x_{11},x_{12}$),($x_{13},x_{14},x_{15},x_{16},x_{17},x_{18}$)
2.  # position and direction of each luminaire
3.  # $IL_1$: all the desk tops; $IL_2$: all the user faces
4.  angle=60
5.  $W_{IL1/DL1}$=Importance($Scene_{IL1},x_1,x_2,x_3,x_4,x_5,x_6$,angle)
6.  $W_{IL1/DL2}$=Importance($Scene_{IL1},x_7,x_8,x_9,x_{10},x_{11},x_{12}$,angle)
7.  $W_{IL1/DL3}$=Importance($Scene_{IL1},x_{13},x_{14},x_{15},x_{16},x_{17},x_{18}$,angle)
8.  $W_{IL2/DL1}$=Importance($Scene_{IL2},x_1,x_2,x_3,x_4,x_5,x_6$,angle)
9.  $W_{IL2/DL2}$=Importance($Scene_{IL2},x_7,x_8,x_9,x_{10},x_{11},x_{12}$,angle)
10. $W_{IL2/DL3}$=Importance($Scene_{IL2},x_{13},x_{14},x_{15},x_{16},x_{17},x_{18}$,angle)
11. $diff_{y12}$=Abs($x_2-x_8$); $diff_{y23}$=Abs($x_8-x_{14}$)
12. $K_0=1/(1+diff_{y12}+diff_{y23})$
13. $K_1=1$; $K_2=100$
14. return $-K_0*(K_1*(W_{IL1/DL1}+ W_{IL1/DL2}+ W_{IL1/DL3})-$
                    $K_2*(W_{IL2/DL1}+ W_{IL2/DL2}+ W_{IL2/DL3}))$
```

Code 2. Cost function

This design study was made with conical projector luminaires with 120º aperture to analyse the effect of few small sized luminaires located asymmetrically in the room: 3 of those luminaires (DLs) were optimised in one side of central vertical panel, although all desk tops and user faces were considered as ILs. The luminaires were also restricted to point downwards, to be above working cells and preferentially aligned in a row. The cost function script is in Code 2. To favour aligned luminaire arrangements, a weighting factor dependent of the Y differences (smaller horizontal dimension) was also included (*lines 11* and *12*). Each set of luminaire parameters was defined with suitable ranges: near the ceiling over each working cell and directed downwards. To effectively prevent glare, the user face IL was scaled to be 100 times bigger than the desk top IL (*line 13*). Fig. 12 (see Appendix) shows the design solution for this setup (geometry of scene, properties of materials, cost function and parameter ranges). Because this study has a complex geometry and many degrees of freedom, each simulation takes 6 hours to complete (2000 iterations). After 200/300 iterations many candidate solutions tend to concentrate in some part of the configuration search space that corresponds to a small set of positions and directions for each DL. Although *Radiance* has an acceleration technique that is very effective, many times radiance distributions are being calculated in points near previously used points (whose radiance distributions can be saved for later reuse). This led us to the development of a new acceleration technique.

5.2.2 Acceleration Technique for SRD Calculation

Whenever a radiance distribution (SRD) has to be calculated in a certain point, it would be useful to reuse available radiance data from neighbouring points, because that could mean a significant reduction in computation time. In Fig. 10 a SRD must be calculated for point A. This technique (see Code 3.) requires a user defined distance threshold (T_{dist}) to decide if a radiance sample from a neighbouring point is used or not – if the perpendicular distance between parallel radiance directions is smaller than T_{dist}, the radiance value is used and no ray-tracing operation is performed. Previously calculated SRDs from points inside the sphere of radius T_{dist} around A are completely used (point B). If necessary, a sorted list of near points is computed and those points are subjected to the T_{dist} check to reuse the available radiance information (for point C in Fig. 10, radiance values from solid angles α_1 and α_2 would be reused in point A). In the end, if there are missing radiance samples, they are computed with ray-tracing operations. To avoid the propagation of radiance val-

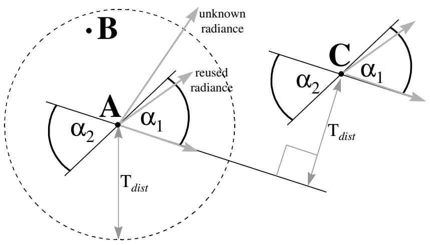

Fig. 10. Acceleration technique

```
calculate_ sum_radiances($P_x,P_y,P_z,D_x,D_y,D_z,A_p,T_{dist}$)
    if SRD($P_x,P_y,P_z,D_x,D_y,D_z,A_p$) already exists
        return weighted sum of radiance values
    initialise this_SRD
    set list_near_SRDS with the nearest SRD's in cache of SRD's
    for each near_SRD in list_near_SRDS
        if radiance values from near_SRD are valid and not reused
            add those radiance values from near_SRD to this_SRD
    if this_SRD is fully computed
        return weighted sum of radiance values
    else
        mark radiance values as reused
    compute missing radiance values using Radiance
    mark those radiance values as new
    add those radiance values to this_SRD
    save this_SRD in the cache of SRD's
    return weighted sum of radiance values
```

Code 3. Acceleration technique

ues from SRD to SRD, which could introduce large radiance errors, each radiance sample carries a flag to indicate if it has already been reused – if so, it is ignored. It is also important to test only a small amount of neighbouring points (less than 30) or else the computation time of this technique becomes greater than the ray-tracing operation. The previous design study was also performed using this acceleration technique with T_{dist} as 5cm and 20 neighbouring points. Design results obtained were very similar (radiance maximum relative error less than 5%). Reduction in computation time was very significant (more than 50%). In a simulation with approximately 6500 iterations, the total amount of radiance calculations was reduced from 18795564 to 5173812 (27.53%). Although this technique requires a considerable amount of bookkeeping, it is not difficult to implement. It produces good results with a significant reduction in computation time even in spaces where desired luminaires are small, very directional and its number is insufficient.

5.2.3 Luminaire Type #2

To overcome the undesirable shadow boundaries generated by spotlight type luminaires, this design study uses 6 omni-directional spherical luminaires with the same design conditions. The lighting design solution is shown in Fig. 13 (see Appendix). The suggested luminaire placement is similar to previous design studies. Although the small size of the spherical luminaires is responsible for some shadow boundaries, ambient lighting is better than in the previous case.

5.2.4 Luminaire Type #3

In this design study an alternative lighting solution was tried to reduce the existence of shadow or penumbra boundaries (mainly due to the small luminaire size). This study uses the original luminaires shown in Fig. 11 (see Appendix), but tries to find an alternative arrangement that optimises the aforementioned lighting design goals. To accomplish this, it was predefined that 5 luminaires would be in a row along the major room dimension, but allowing different spacing distances between them. The design parameters were:

- x, y – reference corner of the leftmost luminaire: over leftmost working cell
- d_1, d_2, d_3, d_4 – spacing between successive luminaires: between 1.5m and 2.5m.

The luminaire height and orientation were also predefined. This originates an optimisation problem with 6 degrees of freedom, but because the luminaires have a large area, each one must be reasonably sampled. We have chosen to sample each luminaire in 4 interior points, which leads to 40 SRD computations per iteration (5 luminaires x 4 points x 2 ILs). Using the reference point of the leftmost luminaire and the 4 spacing distances, the cost function script computes the reference point of the remaining 4 luminaires and samples each at 4 interior points, calculating the contributions of both ILs. Those contributions are finally weighted and combined to produce the cost function value. Fig. 14 (see Appendix) shows an image of the corresponding lighting design solution. It seems to be a good lighting solution because desk top lighting is homogeneous and there are no sharp shadow boundaries visible near the sitting position or any undesirable glare effects. In this study, because the number of SRD calculations per iteration is very big, our acceleration technique produces a significant decrease in computation time. The total amount of radiance calculations for 2300 iterations was reduced from 45258295 to 5147580 (11.37%) and the computation time decreased to less than 30%.

6. Open Questions

Our approach to lighting problems using inverse design seems promising, but there are still many open questions. It is very difficult to convince end users to work with an application based on our methodology and algorithm in its current state. Some of the open questions are discussed next.

6.1 Define lighting design goals

This is a very important question because without a good answer there is no way of solving the design problem. Although it is a difficult question, our experience has shown that most designers are usually trying to solve similar problems, in which the physical component of lighting is the most important and other components (aesthetics, etc) are secondary. Using the IL concept, we think most of the basic lighting design goals can be well described. Lighting design goals related to perceptual aspects, aesthetics, etc, are more difficult to define because they cannot be described by a simple concept (like the IL). Our examples have shown that these goals are more easily defined with some sort of programmable tool like the cost function script we presented. Unfortunately, programming is not a common skill of designers. Some sort of abstraction must be developed to hide this complexity and allow people to clearly and easily state their real lighting design goals.

6.2 What is input data

To solve a lighting design problem, a very basic geometry and little knowledge about the properties of materials is insufficient. Because lighting is closely related to geometry and properties of materials, any design study in which detailed knowledge about those elements is missing will not be able to provide real useful answers. There is even the danger of producing incorrect ones that may completely fail to attain the lighting design goals. This also rises the question of knowing what detail is needed for the geometry and the properties of materials to get correct design solutions in the fastest time possible. Empirically, we can think that small sized objects with mostly diffuse properties could be removed from the geometry, but this must be done carefully and can even lead to serious errors, so it is not generally applicable.

6.3 Find the optimal solution

In our algorithm we are currently using the *ASA* software. Being a simulated annealing algorithm means that it may run for large periods of time until the search is exhausted or something stops its execution. The statistical guarantee of finding the optimal solution is not very useful, because that means that *ASA* finds the optimal solution when it exhausts the search space or with an infinite number of iterations. Our experience with optimisation has shown that veritable solutions are usually found after several thousand iterations (less than 5000), but we cannot ensure that the solution found is optimal or measure its distance from optimality. When the lighting problem context was simple and empirical evidence helped, all solutions found by our algorithm were in agreement with that evidence. But for difficult lighting design problems with complex geometries or complex design goals, empirical evidence is of little help or may even have a negative influence in the solution search. One way to answer this question is by avoiding it. In this case the search would be completely user driven and/or helped by some sort of “intelligent” agents, model based knowledge, genetic algorithms, etc. To accomplish this, the designer should use an interactive graphical interface to test candidate solutions in real time and make decisions on the fly. Unfortunately, this does not seem to be feasible with the existing technology.

6.4 Solve the real problem

Even if it is possible to ensure that quasi-optimal or optimal solutions are generated, there is always the possibility that the solution is not the appropriate one. Errors or uncertainties in the input data (geometry, etc), incorrect specification of goals or other factors may lead to solutions for a problem which is not the designer actual problem. This has to do with the sensitivities between the elements of the problem and the way solutions are generated. For instance, if a highly specular material is described as being only slightly specular, generated solutions may be very different from what they should have been if that specification was correctly made. It seems difficult to measure these sensitivities, but this is an important question because it may affect the designer’s confidence in the generated solution.

6.5 Change input data

Up to now, all our lighting experiments have been made with static geometries of the scene and properties of materials. It would very interesting to solve lighting design problems in which the designer could change small elements in the input data (some small subset of the geometry or some property of a certain material) without having to restart the whole process and discard all previously produced data. This seems to be a very big challenge, but the benefits would be huge and the possibilities fascinating. Probably this will require the development of new lighting algorithms which are able to separate and classify meticulously all the lighting exchanges so that partial recalculations become possible without loosing accuracy.

6.6 Future work

In the future we will try to address and give answers to some of the simplest questions raised. The large computation times are a consequence of the simplicity of our algorithm and methods/techniques chosen, so further research has to be made to replace them with more efficient alternatives. The optimisation part of our algorithm also needs to be improved so that quasi-optimal solutions are produced as soon as possible and the amount of iterations performed to produce them decreases significantly. If improved, these two factors will have a great influence in reducing the computation times. The main goal of a project named CASSILDE (*Computer ASSIsted Lighting DEsign*), funded by portuguese ministry of Science and Technology, is the development of an easy to use and intuitive prototype application for designers based on this research work.

7. Conclusions

The main objective of our work is to develop new approaches that can solve the lighting design problem while trying to improve and shorten the design cycle. This paper explains in detail some developments in that approach. It has originated a methodology and an algorithm that semi-automatically tries to solve the lighting design problem. Our algorithm uses well-known tools from Global Illumination and Optimisation fields and links them in a new innovative way. We also present a new acceleration technique for radiance calculations within our algorithm. It allows a faster calculation of radiance distributions and seems to be very effective in reducing the computation times (50-75% reductions in all tested cases). The examples presented in this paper are complex and show the possibilities of this new approach. The solutions found for the several examples seem consistent with empirical evidence, feasible and veritable. In this paper we also present some important open questions related to lighting design and discuss some ideas and possible answers.

References

[Cohen93] Cohen, M.F.; Wallace, J.R.; *Radiosity and Realistic Image Synthesis*; Academic Press 1993.

[Costa99] Costa, A.; Sousa, A.; Ferreira, F.; *Optimisation and Lighting Design - WSCG'99 Proceedings*, Short Papers, pages 29-36; WSCG 1999.

[Feynm85] Feynman, R.; *QED: The Strange Theory of Light and Matter*, Princeton University Press 1985.

[Inbge93] Ingber, L.; *Simulated Annealing: Practice versus Theory - Journal of Mathematical Computer Modelling*, V. 18, N. 11, pages 29-57; 1993.

[Ingbe98] Ingber, L.; *Adaptive Simulated Annealing*; *ftp://ftp.ingber.com/pub*; Lester Ingber Research; 1993-1998.

[Kawai93] Kawai, J.K.; Painter, J.S.; Cohen, M.F.; *Radioptimization - Goal Based Rendering - COMPUTER GRAPHICS Proceedings*, pages 147-154; SIGGRAPH 1993.

[Kirkp84] Kirkpatrick, S; *Journal of Statistical Physics*, V. 34, pages 975-986; 1984.

[Marks97] Marks, J; Andalman, B.; Beardlsey, P.A.; Freeman, W.; Gibson, S.; Hodgins, J.; Kang, T.; *Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation - COMPUTER GRAPHICS Proceedings*, pages 389-400; SIGGRAPH 1997.

[Pirlo96] Pirlot, M.; *General Local Search Methods - European Journal of Operational Research*, N. 92, pages 493-511; Elsevier Science B.V. 1996.

[Pouli97] Poulin, P.; Ratib, K.; Jacques, M.; *Sketching Shadows and Highlights to Position Lights - Proceedings of Computer Graphics International 97*, pages 56-63, 1997.

[Schoe93] Schoeneman, C.; Dorsey, J.; Smits, B.; Arvo, J.; Greenberg, D.; *Painting with Light - COMPUTER GRAPHICS Proceedings*, pages 143-146; SIGGRAPH 1993.

[Veach97] Veach, E.; *Robust Monte Carlo Methods for Light Transport Simulation*, PhD Thesis, Chapter 7, pages 201-218, Stanford University, 1997.

[Ward94] Ward, G.J.; *The Radiance Lighting Simulation and Rendering System - COMPUTER GRAPHICS Proceedings*, pages 459-472; SIGGRAPH 1994.

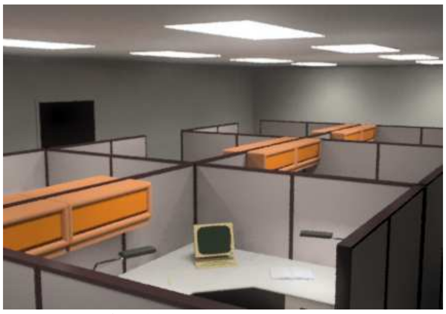

Fig. 11. Empirical lighting design solution

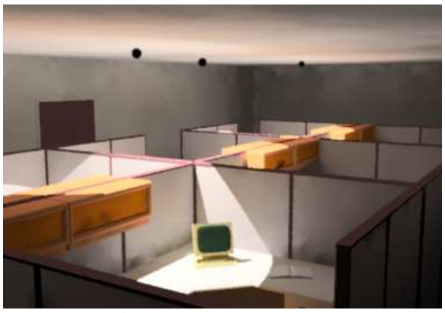

Fig. 12. Lighting design solution for 120º DLs (3)

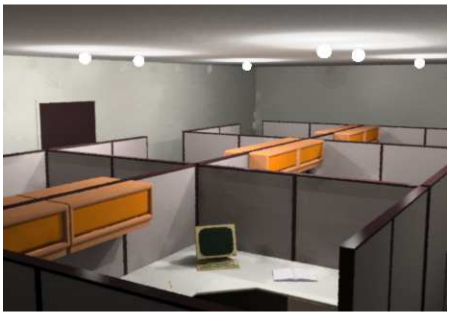

Fig. 13. Lighting design solution for 360º DLs (6)

Fig. 14. Lighting design solution for predefined DLs (5)